Quiz 1

Use the data set given in the data file exam1.txt to build a regression model by implementing the cost function and gradient descent algorithm where the cost function in regression is

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Then plot the relation between cost history and number of iterations.

Quiz 2

Build logistic regression model using **exam2.txt** by implementing the following tasks:

1. Load the data and display it on a 2-dimensional plot.
2. Implement function to map the features
3. Implement the cost function and gradient descent algorithm where the cost function in regression is

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right],$$

and the gradient of the cost is a vector of the same length as θ where the j^{th} element (for $j = 0, 1, \ldots, n$) is defined as follows:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

4. Plot the decision boundary using the given function

```
function plotDecisionBoundary(theta, X, y)

    plotData(X(:,2:3), y);
    hold on
    if size(X, 2) <= 3
    % Only need 2 points to define a line, so choose two endpoints
        plot_x = [min(X(:,2))-2,  max(X(:,2))+2];
        plot_y = (-1./theta(3)).*(theta(2).*plot_x + theta(1));
        plot(plot_x, plot_y)
    else
        % Here is the grid range
        u = linspace(-1, 1.5, 50);
        v = linspace(-1, 1.5, 50);
        z = zeros(length(u), length(v));
        % Evaluate z = theta*x over the grid
        for i = 1:length(u)
            for j = 1:length(v)
                z(i,j) = mapFeature(u(i), v(j))*theta;
            end
        end
        z = z'; % important to transpose z before calling contour
```

```
            % Plot z = 0
            % Notice you need to specify the range [0, 0]
            contour(u, v, z, [0, 0], 'LineWidth', 2)
        end
        hold off

end
```

Quiz 3

Build regularized logistic regression model using **exam2.txt** by implementing the following tasks:

1. Load the data and display it on a 2-dimensional plot.
2. Implement function to map the features using the given function

```
function out = mapFeature(X1, X2,degree)
out = ones(size(X1(:,1)));
for i = 1:degree
    for j = 0:i
        out(:, end+1) = (X1.^(i-j)).*(X2.^j);
    end
end

end
```

3. Implement the cost function and gradient descent algorithm where the cost function in regression is

$$J(\theta) = \left[-\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log \left(h_\theta(x^{(i)}) + (1 - y^{(i)}) \log 1 - h_\theta(x^{(i)}) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \right.$$

4. Plot the decision boundary using the given function

```
function plotDecisionBoundary(theta, X, y)

    plotData(X(:,2:3), y);
    hold on
    if size(X, 2) <= 3
   % Only need 2 points to define a line, so choose two endpoints
            plot_x = [min(X(:,2))-2,  max(X(:,2))+2];
            plot_y = (-1./theta(3)).*(theta(2).*plot_x + theta(1));
            plot(plot_x, plot_y)
    else
            % Here is the grid range
            u = linspace(-1, 1.5, 50);
            v = linspace(-1, 1.5, 50);
            z = zeros(length(u), length(v));
            % Evaluate z = theta*x over the grid
            for i = 1:length(u)
                    for j = 1:length(v)
                            z(i,j) = mapFeature(u(i), v(j))*theta;
                    end
            end
```

```
            z = z'; % important to transpose z before calling contour

            % Plot z = 0
            % Notice you need to specify the range [0, 0]
            contour(u, v, z, [0, 0], 'LineWidth', 2)
        end
        hold off

end
```